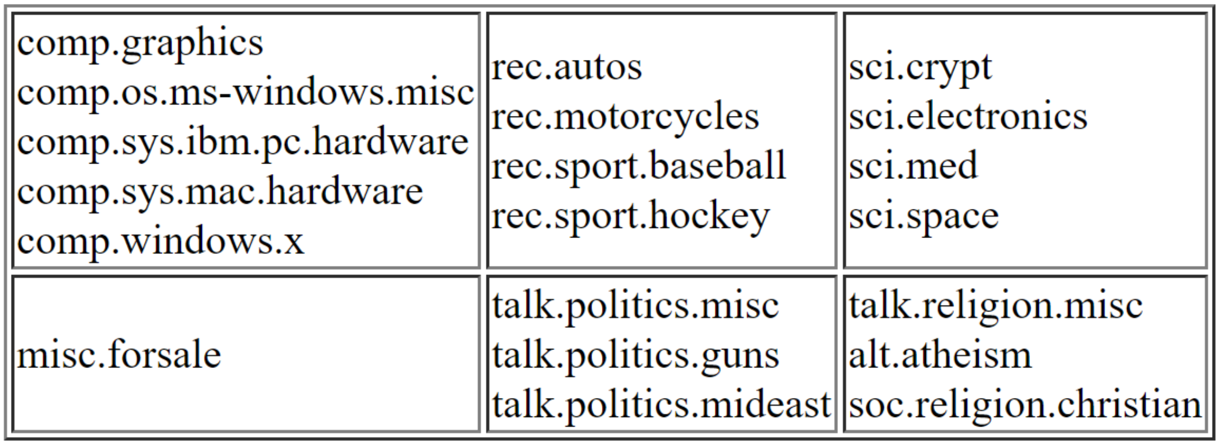
Project 2

About the Dataset:

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. To the best of my knowledge, it was originally collected by Ken Lang, probably for his *Newsweeder: Learning to filter netnews* paper, though he does not explicitly mention this collection. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering[1].

Here is a list of the 20 newsgroups, partitioned (more or less) according to subject matter:



comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Naive Bayes Classifier:

In machine learning, **naïve Bayes classifiers** are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models[2].

The Bayes formula is as follows:

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}$$

- P(A) is the prior probability of A occuring independantly.
- P(B) is the prior probability of B occuring independantly.
- P(A|B) is the posterior probability that A occurs given B.
- P(B|A) is the likelihood probability of B occuring, given A.

It is a popular method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features[2].

Data Preprocessing:

I am dividing the each dataset file into training(500) and testing set(500). I am using nltk to download the stop words and remove then as well as punctation marks from the each file. It replaces the punctation marks with spaces and tokenizes the word and increasing the counter each time a new word is found during preprocessing.

Training the Dataset:

After Data preprocessing, we train the classifier such that if it encounters new word in the dataset it checks the dictionary and add it if it is unique. We find the probability of each class and a word in each class.

Testing the Dataset:

After the training, we check the testing dataset against the dictionary of words we got after training the dataset. We use stemmer which reduces the word to the root/base word. We find the probability of the each subclass occurring given that the probability of the each subclass training data. We find the accuracy, error and execution time.

Structure of the Code

1) First download the data set from the given URL.
2) I am creating a dictionary to store the value of the dataset,
3) Then I split the data into train and test datasets (i.e) the first 500 will be in the train and the remaining 500 will be on the test data.
4) With the given data we have to clean the data by removing unnecessary words by using various methods like set(which gives the unique words in the list) and other functions
5) Then we find the probability of each class and a word in each classes.
6) With the given probabilities we implement the Naive Bayes classifier and find the accuracy which is the avg of all the classes.

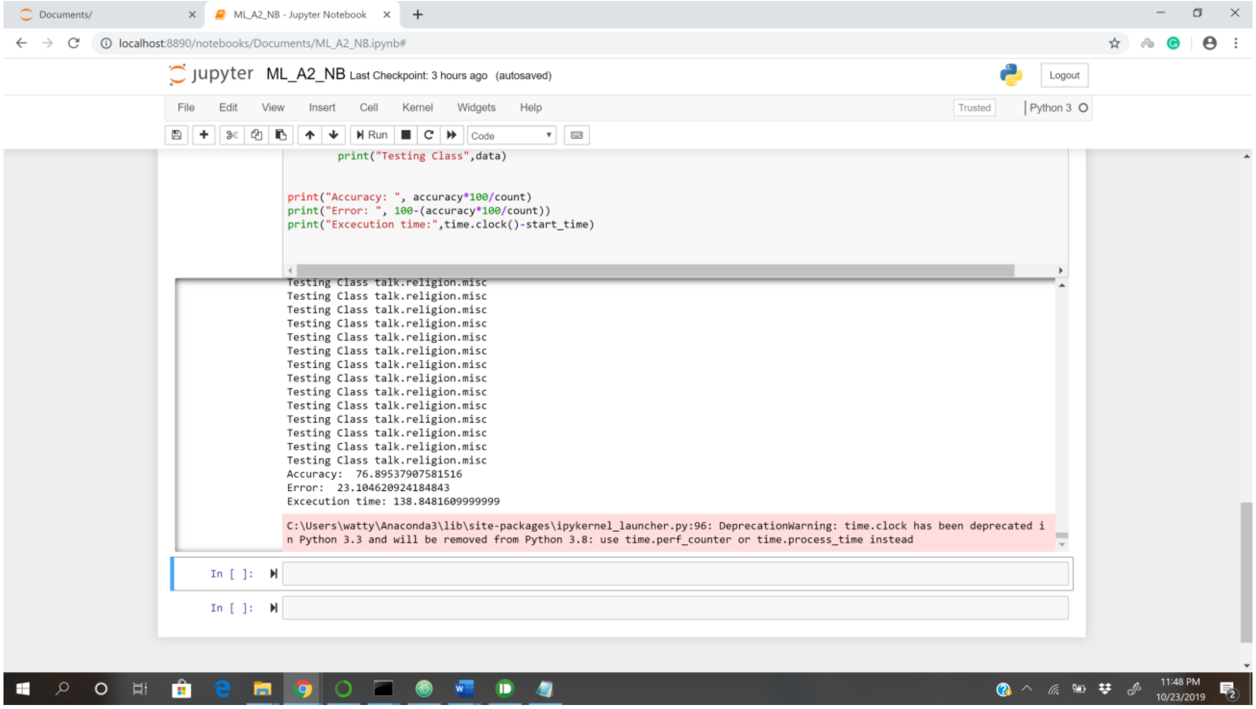
Result:

From the given data the accuracy, the error and the execution time I got is,

Accuracy: 76.89%

Error: 23.10%

Execution time: 138.84 seconds


Documents/
ML_A2_NB - Jupyter Notebook
localhost:8890/notebooks/Documents/ML_A2_NB.ipynb#
jupyter ML_A2_NB Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 ○
Run
Code
In [ ]:
In [ ]:
11:48 PM
10/23/2019

```
        print("Testing Class",data)


print("Accuracy: ", accuracy*100/count)
print("Error: ", 100-(accuracy*100/count))
print("Excecution time:",time.clock()-start_time)
```

```
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Testing Class talk.religion.misc
Accuracy:  76.89537907581516
Error:  23.104620924184843
Excecution time: 138.8481609999999

C:\Users\watty\Anaconda3\lib\site-packages\ipykernel_launcher.py:96: DeprecationWarning: time.clock has been deprecated i
n Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead
```

References

1) http://qwone.com/~jason/20Newsgroups/
2) https://en.wikipedia.org/wiki/Naive_Bayes_classifier